Telemetry in YARP

Yes or No?

Working group on Logging/Telemetry meeting

Daniele E. Domenichelli
daniele.domenichelli@iit.it
Istituto Italiano di Tecnologia (IIT), HSP

2020-10-14

ISTITUTO ITALIANO
DI TECNOLOGIA

Telemetry

- ✓ Is Telemetry something that we would like to have in YARP?
  - Useful for the all the YARP user?
  - Does it require lots of work?
  - Can it give visibility to YARP and/or to the new telemetry library?
  - Does the library make sense in a context without YARP?
  - Do we want to use this library in YARP?
  - Should YARP depend on this library? Or should this library depend on YARP?
  - Do we want to add an extra component/library, whose funcionality partially overlap with YARP?
  - Politics (HSP vs iCub Tech), release schedules

Telemetry

- ✓ Can be split in 3 different “steps”
  a. **publishing** (add)
  b. **filtering** (from config file and eventually environment variables)
  c. **esporting** (using matio, stdout, network, etc. - Perhaps using plugins)

Telemetry

- ✓ What is overlapping with YARP?
  - Partially overlapping with YARP ports.
  - Partially overlapping with YARP logger and log components.
  - “Snapshot” of the state of a component.
    - All the state is saved by the telemetry.
    - A part of the state is published on YARP “stream” ports.
    - Parts of the state can be requested using YARP “rpc” ports.
    - A part of the state can be logged using YARP logger.
  - Filtering is very similar to the “LogComponent” system in YARP.
    - Configuration is not there yet, but it is exactly the same thing.

Development Model

- ✓ In a separate (non-YARP) library?
- ✓ In `yarp::os`?
- ✓ In `yarp::telemetry` in YARP repository?
- ✓ In `yarp::telemetry` in a separate repository?

Development Model

- ✓ In a separate (non-YARP) library:
  - Pros:
    - Faster development
    - API not necessarily stable
  - Cons:
    - Less visibility
    - Extra dependency for all the devices using it
    - Cannot be used in YARP devices or adds an extra dependency
    - Partial overlapping with YARP functionalities.

Development Model

- ✓ In `yarp::os`?
  - Pros:
    - Can be used in YARP (including in `yarp::os`)
    - I’m going to thoroughly review every single change.
    - Possibly reuse parts of code.
    - Optionally telemetry inside port for published data (no need to add and `write`)
    - No circular dependencies
  - Cons:
    - Slower development
    - Strict API stability requirement.
    - I’m going to thoroughly review every single change.

Development Model

- ✓ In `yarp::telemetry` in YARP repository?
  - Pros:
    - Can be used in YARP devices and optionally also in `yarp::os` (but in that case it cannot use anything from `yarp::os` to avoid circular dependencies).
    - Can reuse parts of YARP code.
    - I'm going to review the changes.
    - Start as `yarp::telemetry::experimental`
  - Cons:
    - Slower development
    - I'm going to review the changes.

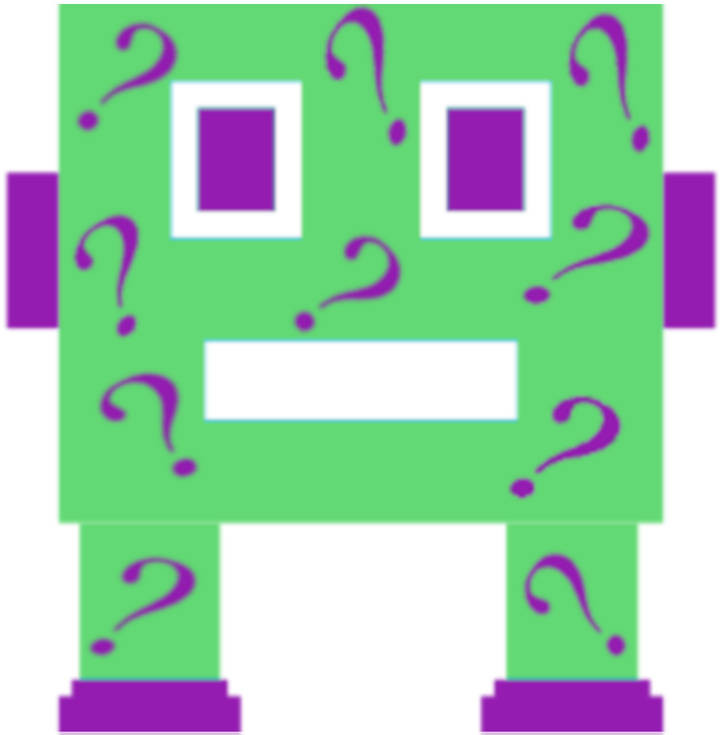
Development Model

- ✓ In `yarp::telemetry` in a separate repository?
  - Almost the same as developing inside YARP
  - Pros:
    - Faster development.
    - Can be moved later in YARP.
    - Cannot be used in YARP, but can be easily moved in YARP later.
    - If we realize we need it without YARP, it can be easily reverted to a standard library.
  - Cons:
    - ?

Development Model

- ✓ In a separate (non-YARP) library? (maybe)
- ✓ In `yarp::os`? (maybe)
- ✓ In `yarp::telemetry` in YARP repository? (maybe)
- ✓ In `yarp::telemetry` in a separate repository? (IMHO best option for starting)

Let’s Discuss!



https://telegram.me/addstickers/YARPino